**Que signifie le terme NoSQL :**

Not Only SQL

Def => concept et un modèle de base de données flexible. En général et spécifiquement NoSQL ne suit pas les règles d’une base de données relationnelle (RDBMS). NoSQL n’utilise pas non plus le langage de requête SQL. NoSQL est un modèle de base de données différent de SQL.

**Comment sont nées les bases NoSQL :**

Le terme NoSQL a été publié pour la première fois par Carlo Strozzi en 1998 pour nommer la base de données qu’il développait à l’époque, « Strozzi NoSQL open-source relational database ». Strozzi a déclaré que la façon dont NoSQL fonctionnait à cette époque était « très différente » des bases de données relationnelles dans leur ensemble, c’est la raison pour laquelle il l’a appelée « NoREL », qui fait référence au terme « No Relational ».

Fin 2000, le développement de NoSQL a repris, dans le but de dépasser les limites de SQL, notamment en termes de scalabilité et de potentiel de collecte de données multi-structurées.

Début 2009, Johan Oskarsson, l’un des développeurs de Last.fm, a réintroduit le terme NoSQL lorsqu’il a organisé un événement pour discuter des « bases de données distribuées non relationnelles open source ».

Les bases de données NoSQL ont été développées dans l’espoir de résoudre les problèmes classiques des bases de données SQL. Les bases de données NoSQL sont différentes des bases de données SQL en ce sens qu’il n’y a qu’un seul type de méthode de stockage des données. La structure de la base de données NoSQL est plus dynamique et flexible avec quatre (4) types de modèles de stockage de données, qui sont les suivants, accompagnés d’exemples et de brèves définitions.

Aujourd’hui :

Alors que les entreprises et les organisations ont besoin d'innover rapidement, il est essentiel pour elles de pouvoir rester agiles et continuer à travailler à n'importe quelle échelle. Les bases de données NoSQL proposent des schémas flexibles et prennent également en charge une variété de modèles de données idéaux pour créer des applications qui nécessitent de grands volumes de données et des temps de latence ou de réponse faibles (par exemple, des jeux en ligne et des applications Web d'e-commerce).

**Pourquoi parle-t-on de « non relationnel » :**

 Il ne nécessite pas de schéma et n’a pas de relations de table

**Y-a-t-il un seul type de base NoSQL :**

La structure de la base de données NoSQL est plus dynamique et flexible avec quatre (4) types de modèles de stockage de données, qui sont les suivants, accompagnés d’exemples et de brèves définitions.

1. **Magasin clé-valeur (KV)** : ArangoDB, Apache Ignite, Oracle NoSQL Database, Couchbase, Dynamo, Redis, Riak.  
   Le stockage KV utilise un tableau associatif, également appelé carte ou dictionnaire, comme base de son modèle de données. Dans ce modèle, les données sont présentées sous la forme d’une collection de CV appariés.
2. **Basé sur des documents** : Apache CouchDB, ArangoDB, BaseX, Clusterpoint, Couchbase, Cosmos DB, IBM Domino, MarkLogic, MongoDB, OrientDB, Qizx, RethinkDB.  
   Le concept de stockage de données basé sur des documents est une idée du concept de « document ». Où chaque implémentation de base de données basée sur des documents est généralement supposée que les documents sont encapsulés et codés dans un format de codage standard.
3. **Basé sur des colonnes** : Accumulo, Cassandra, Scylla, HBase.  
   Également connu sous le nom de modèle à colonnes larges, il permet un accès très rapide aux données en utilisant des clés de ligne, des noms de colonne et des horodatages de cellule. La souplesse du schéma de ce type de base de données permet aux enregistrements dans ses colonnes (colonnes) de ne pas être cohérents. Les utilisateurs peuvent ajouter des colonnes dans une ligne spécifique sans avoir à les ajouter dans chaque enregistrement.
4. **Basé sur les** graphes : AllegroGraph, ArangoDB, InfiniteGraph, Apache Giraph, MarkLogic, Neo4J, OrientDB, Virtuoso.  
   En théorie, les structures basées sur des graphes sont constituées de sommets et d’arêtes (données et connexions) qui peuvent être appelées relations de données. Les graphes se comportent de la même manière que les humains pensent, les données sont disposées dans des relations spécifiques entre les unités de données qui ont leurs propres caractéristiques. Ce type de base de données est utile pour visualiser, analyser et vous aider à trouver des connexions entre différentes données.

**Dans quel cas utiliser une base NoSQL :**

Dans certains cas, NoSQL est considéré comme fonctionnant mieux que le SQL relationnel, notamment les suivants.

1. Lorsque vous devez stocker de grandes quantités de données avec des schémas incohérents.  
   Le schéma de données dans NoSQL n’est pas  *fixe* comme dans SQL, les modifications de la structure et du schéma qui peuvent changer à tout moment peuvent être gérées facilement sans avoir à modifier la cohérence des données dans la base de données elle-même.
2. Lorsque vous avez besoin  d’une informatique et d’un stockage de données *basés sur le cloud  .*  
   La plupart des bases de données NoSQL sont construites et conçues pour fonctionner dans différents  *centres de données* et fonctionner comme des systèmes distribués. Dans ce cas, vous en bénéficierez en tant qu’utilisateur NoSQL, car vous pourrez profiter de l’accès à diverses infrastructures informatiques basées sur le cloud.
3. Lorsque vous êtes un *développeur Web*ou un développeur d’*applications et que vous avez besoin d’une mise à jour* rapide (  *rapide* )  . Avec NoSQL, vous n’avez pas besoin de préparer les données comme si vous utilisiez RDBMS (SQL), vous pouvez même migrer des données structurées d’une version de l’application à la prochaine version mise à jour, chaque fois que la conception de l’application est mise à jour. La nature dynamique de NoSQL peut évoluer à mesure que l’application change.

**Les inconvénients du NoSQL :**

L’inconvénient **majeur** des bases de données NoSQL est l’absence de règles de standardisation. En effet, les données sont **dé-normalisées** et traitent avec des enregistrements (Documents par exemple pour le type de données document) sur les serveurs contrairement au SQL qui marche par **référencement**.

C’est un problème car beaucoup d’utilisateurs préfèrent des données **normalisées**. Celles-ci permettent d’éviter les **anomalies** et les **duplications** de données.

Par exemple, les**transactions financières** utilisent des règles de normalisations très **strictes**. Utiliser des données non normalisées peut engendrer un risque d’erreur dans les transactions (une anomalie) qui pourrait avoir un **impact très grave** pour les utilisateurs.

Ensuite, le NoSQL ne s’associe pas avec une utilisation **complexe** des requêtes. Il fonctionne très bien sur une table mais pour manipuler plusieurs tables, il est vite **limité**. En effet, le NoSQL ne peut pas faire de **jointures complexes** (qui sont nécessaires pour manipuler plusieurs tables). De plus, il ne peut pas imbriquer les requêtes WHERE comme le fait le **SQL** pour extraire des informations très spécifiques dans la database.